Multi-robot Learning and Coverage of Unknown Spatial Fields

María Santos[1], Udari Madhushani[1], Alessia Benevento[2], and Naomi Ehrich Leonard[1]

Abstract— **This paper addresses the problem of optimally covering a domain when the scalar function that describes the relative importance of the points in the domain is initially unknown. We propose an adaptive strategy for a team of cooperative robots that combines estimation and learning methods with optimal spatial coverage. The proposed algorithm leads the team of robots to an optimal solution of the coverage problem by efficiently trading off movement choices for learning the field with movement choices for covering the estimated field. The algorithm exploits the flexibility of Gaussian processes for learning the field and optimization rules based on Voronoi partitions of the environment for covering the field. We propose an exploration strategy that uses the decentralized nature of the coverage problem by allowing each robot to sample the space in its area of dominance. We provide a theoretical guarantee of the algorithm. The performance of the proposed algorithm is evaluated in simulation as well as on a team of mobile robots.**

I. INTRODUCTION

The spatial coverage problem refers to how to position a collection of mobile robots over a domain such that its environmental features are optimally monitored [1], [2], [3], [4]. Coverage algorithms offer strategies for individual robots to distribute themselves over the environment, affording higher concentrations of robots in more important areas. This makes the control paradigm attractive for applications such as search and rescue [5], precision agriculture [6], and environmental monitoring and exploration [7], [8], [9], [10].

When covering a domain, it is likely that not all the points require the same attention from the multi-robot team. Locations with high concentration of resources, prevalence of environmental features, or increased probability associated with certain events, require closer attention from the robots. Variability of the concentration, prevalence or probability can be encoded as a *density function* defined over the domain, as in [1]. Different coverage control approaches have addressed a variety of scenarios associated with these density functions, including the homogeneous, static case [1], [11], heterogeneous densities associated with different sensing capabilities [12], [13], nonuniform fields [14], and time-varying densities that represent dynamic objectives [15], [16]. However, a common assumption is to consider the density function as known information to each of the robots tasked with covering the environment.

This work was supported by the Office of Naval Research grant 25105-G0001-10012165 and Army Research Office grant W911NF-18-1-0325.

[1]M. Santos, U. Madhushani, and N. E. Leonard are with the Department of Mechanical and Aerospace Engineering, Princeton University, Princeton, NJ, 08544 USA; maria.santos@princeton.edu, udarim@princeton.edu, naomi@princeton.edu

[2]A. Benevento is with the Department of Engineering, University of Salento, via per Monteroni, 73100, Lecce, Italy; alessia.benevento@unisalento.it

In this work, we focus on how coverage should be performed when the density function over the domain is initially unknown to the multi-robot team. In the proposed algorithm, each robot explores the domain and sequentially collects samples of the density function, which are used to adaptively learn the function even in the unexplored areas. The learning task is performed simultaneously with the coverage task: we propose an optimization problem in which the cost function is a combination of the learning goal and the coverage goal. As more data is added to the model, the approximation of the density function becomes more accurate and the learning task is de-emphasized in favor of the coverage task. The learning task is performed using a Bayesian approach that leverages the flexibility of Gaussian processes. The estimation is made more accurate by having each robot sample the space in its area of dominance so as to maximize its knowledge of the area and to preserve the decentralized nature of the problem. The coverage task is achieved by leveraging the well-known Lloyd's algorithm [17]. The proposed method is theoretically justified by exploiting existing results on regret analysis of Bayesian optimization problems.

The paper is organized as follows. The rest of this section includes a summary of related literature on multi-robot estimation and spatial coverage. Section II formally introduces the spatial coverage problem and the considerations associated with sampling the density function. In Section III we describe the probabilistic model used to learn the density function from the robot samples. The proposed algorithms are described and justified in Section IV, and their performance is analyzed in simulation and on real robotic platforms in Section V. We conclude in Section VI.

A. Related Work

A broad body of work has been dedicated to the problem of multi-robot optimal coverage, e.g., [1], [18], and the problem of cooperative exploration and estimation of unknown fields (often for coordinated control), e.g., [19], [20], [21], [22], [23]. For a comprehensive review see [24]. However, the question of how to optimally explore a spatial field to learn it *and* optimally cover it remains a problem under investigation.

Some works optimize only one or the other problem. For example, in [25], [26] the robots move to optimally cover the estimated field, which is updated from the measurements taken along the way. No optimal exploration strategy is defined. Conversely, in [8], [27], an optimal exploration strategy is defined to minimize uncertainy in the estimated field, but no strategy to optimally cover the sampled field is defined. Other works use a sequential, phased execution

of both tasks, e.g., [28], [29], [30], [31]. Typically, the first phase involves an optimal exploration of the domain to estimate the density function. In the second phase, the estimate is used by the team to optimally cover the environment.

The algorithm proposed in [32] does address the optimal estimation and optimal coverage tasks simultaneously. However, the convergence of the algorithm requires the use of an agent dedicated to sampling points in the domain for learning while the other agents perform optimal coverage without focusing on decreasing the uncertainty of the estimation. In contrast, we propose two sampling strategies that enable the multi-robot team to optimally learn the spatial field and optimally cover it.

II. SPATIAL COVERAGE WITH AN UNKNOWN DENSITY FUNCTION

Coverage control deals with the problem of how to distribute a team of robots over a domain in order to optimally monitor its environmental features. Consider a team of N mobile robots with positions $x_i \in \mathbb{R}^n$, $i \in \{1, \dots, N\}$, that are to cover a compact, convex domain, $D \in \mathbb{R}^n$. To distribute the covering responsibilities among the team, a typical choice is to let each Robot i be in charge of those points that are closest to it, i.e., in its region of dominance

$$V_i(\mathbf{x}) = \{q \in D \mid \|q - x_i\| \leq \|q - x_j\|,\ \forall j \neq i\},$$

where $\mathbf{x} = [x_1^\intercal, \dots, x_N^\intercal]^\intercal \in \mathbb{R}^{Nn}$ is the stacked position of the robots. The region of dominance of each robot thus becomes its Voronoi cell in the domain D with respect to the Euclidean distance, which can be calculated by each robot by measuring the relative position of its neighbors and the environment boundaries.

The prevalence of features of interest may vary across the domain. The relative importance of the different points in the domain is typically represented through a spatial field or density function, $\phi : D \mapsto [0, \infty)$, that encodes the relative importance of the points in the domain such that, the higher the value of the density $\phi(q)$ at a particular point, the more attention we want the team to pay to that point. A common assumption in the coverage literature (e.g., [1], [2], [33]) is for this density function to be known by the team. With this information, the quality of coverage of Robot i over its region of dominance $V_i(\mathbf{x})$ can be encoded as

$$h_i(\mathbf{x}, \phi) = \int_{V_i(\mathbf{x})} \|q - x_i\|^2 \phi(q)\, dq,$$

where the square of the Euclidean distance between the position of the robot and the points within its region of dominance reflects the degradation of the sensing performance with distance. Summing over all the agents, the performance of the multi-robot team with respect to ϕ can then be encoded through the locational cost [1] as

$$\mathcal{H}(\mathbf{x}, \phi) = \sum_{i=1}^{N} h_i(\mathbf{x}, \phi) = \sum_{i=1}^{N} \int_{V_i(\mathbf{x})} \|q - x_i\|^2 \phi(q)\, dq, \quad (1)$$

where the lower the cost, the better the coverage.

A necessary condition for (1) to be minimized is that the position of each robot corresponds to the center of mass of its Voronoi cell [33], given by

$$c_i(\mathbf{x}) = \frac{\int_{V_i(\mathbf{x})} q\phi(q)\, dq}{\int_{V_i(\mathbf{x})} \phi(q)\, dq}. \quad (2)$$

This spatial configuration, referred to as a centroidal Voronoi tessellation (CVT), can be achieved by letting the multi-robot team execute the well-known Lloyd's algorithm [17]:

$$\dot{x}_i = \kappa(c_i(\mathbf{x}) - x_i), \quad \kappa > 0. \quad (3)$$

However, CVTs are not unique and, thus, (3) only guarantees convergence to a local minimum of (1). Furthermore, note that $c_i(\mathbf{x})$ only depends on the position of the Delaunay neighbors of Robot i, i.e., the set of robots whose Voronoi cells share a face with Robot i's Voronoi cell.

In this work, however, we assume that the density function ϕ is unknown. Robots can take a scalar noisy measurements of the density function as they move around the domain,

$$y_i = \phi(x_i) + \varepsilon, \quad (4)$$

where $\varepsilon \sim \mathcal{N}(0, \sigma^2)$. The measurement noise ε is independent and identically distributed (i.i.d.) across time and space.

For the remainder of the paper, we use the following notation. T is the total number of time iterations and $t \in \{1, \dots, T\}$ denotes the time iteration. The set of points sampled by all the robots in the domain D at time t is denoted by $\mathbf{x}^{(t)} = \left[x_1^{(t)\intercal}, \dots, x_N^{(t)\intercal}\right]^\intercal \in \mathbb{R}^{Nn}$. The corresponding noisy measurements are $\mathbf{y}^{(t)} = [y_1^{(t)}, \cdots, y_N^{(t)}]^\intercal \in \mathbb{R}^N$. We let $\mathbf{x}^{(1:t)}$ and $\mathbf{y}^{(1:t)}$ refer to all the sampled locations and noisy measurements taken from iteration 1 to t.

At each time t, all the sampled information taken by the robots from the beginning of the algorithm, $\{\mathbf{x}^{(1:t)}, \mathbf{y}^{(1:t)}\}$, can be used to estimate the density function over the domain. In the next section, we introduce the Gaussian process and Bayesian inference tools used in this paper.

III. GAUSSIAN PROCESSES FOR FIELD ESTIMATION

A challenging assumption considered in this work is that the density function $\phi(\cdot)$ in equation (1) is not known in advance and must be learned online. For this reason, we assume that robots are equipped with an estimation mechanism that progressively refines their knowledge of the density function. In this section, we address the problem of learning $\phi(\cdot)$ from points sampled by the robots. To this end, we employ Gaussian processes. A Gaussian process is a generalization of the Gaussian probability distribution, and it can represent a function explicitly and rigorously by placing a prior distribution over the space of functions. The combination of the prior and the data leads to the posterior distribution over functions.

A Gaussian process is completely specified by a mean function $\mu : D \to \mathbb{R}$ and covariance function $k : D \times D \to \mathbb{R}$, where μ represents the expected value of ϕ at point x, and k measures the linear dependence of two variables x and x'

[34]. We assume that the structures of μ and k are known up to certain hyperparameters ρ and τ as

$$\mu(\cdot,\rho), \qquad k=(\cdot,\cdot,\tau).$$

In particular, in Section VI we use a linear model for the mean and a squared-exponential kernel for the covariance,

$$\mu(x;\rho)=\rho^{\intercal}x, \qquad k(x,x';\tau)=\exp\left(-\frac{\|x-x'\|^2}{2\tau^2}\right).$$

Thus, the unknown density function can be described with a Gaussian process,

$$\phi(\cdot)\sim GP(\mu(\cdot,\rho),k(\cdot,\cdot,\tau)).$$

The unknown hyperparameters can be estimated from the data through maximum likelihood estimation. Once the hyperparameters are estimated, Bayesian inference is used to model an estimation of ϕ. The estimation task in the algorithm is addressed as follows.

At each time $t\in\{1,\dots,T\}$ the collected samples $\mathbf{y}^{(1:t)}$ are used to perform maximum likelihood estimation of the hyperparameters as

$$(\rho^{(t)},\tau^{(t)})=\underset{\rho,\tau}{\operatorname{argmax}}\, p(\mathbf{y}^{(1:t)}\mid\mathbf{x}^{(1:t)};\rho,\tau). \tag{5}$$

The function p of equation (5) can be expressed in closed form with log-likelihood function:

$$\begin{aligned}&\log(p(\mathbf{y}^{(1:t)}|\mathbf{x}^{(1:t)};\rho,\tau))=\\ &-\frac{1}{2}(\boldsymbol{\mu}^{(1:t)}(\rho)-\mathbf{y}^{(1:t)})^{\intercal}(\mathbf{K}_{\mathbf{x}^{(1:t)}}(\tau)+\sigma^2\mathbf{I})^{-1}(\boldsymbol{\mu}^{(1:t)}(\rho)-\mathbf{y}^{(1:t)})\\ &\qquad-\frac{1}{2}\log|\mathbf{K}_{\mathbf{x}^{(1:t)}}(\tau)+\sigma^2\mathbf{I}|-\frac{t}{2}\log(2\pi),\end{aligned}$$

where $\boldsymbol{\mu}^{(1:t)}(\rho)\in\mathbb{R}^{Nt}$ is the vector of mean values at all points sampled

$$\boldsymbol{\mu}^{(1:t)}(\rho)\triangleq[\mu(x_1^{(1)};\rho)\cdots\mu(x_n^{(t)};\rho)]^{\intercal},$$

$\mathbf{I}$ is the identity matrix, and $\mathbf{K}_{\mathbf{x}^{(1:t)}}(\tau)\in\mathbb{R}^{Nt\times Nt}$ is the covariance matrix with entries $k(x_i^{(j)},x_{i'}^{(j')};\tau)$ for $i,i'\in\{1,\dots,N\}$ and $j,j'\in\{1,\dots,t\}$. Therefore, the maximum likelihood estimation of ρ and τ is given by

$$\begin{aligned}&(\rho^{(t)},\tau^{(t)})=\underset{\rho,\tau}{\operatorname{argmin}}\big\{\log|\mathbf{K}_{\mathbf{x}^{(1:t)}}(\tau)+\sigma^2\mathbf{I}|+\\ &(\boldsymbol{\mu}^{(1:t)}(\rho)-\mathbf{y}^{(1:t)})^{\intercal}(\mathbf{K}_{\mathbf{x}^{(1:t)}}(\tau)+\sigma^2\mathbf{I})^{-1}(\boldsymbol{\mu}^{(1:t)}(\rho)-\mathbf{y}^{(1:t)})\big\}\end{aligned} \tag{6}$$

The maximum likelihood solution of the hyperparameters (6) improves as more samples are added.

Given hyperparameters $\rho^{(t)}$ and $\tau^{(t)}$, noisy observations $\mathbf{y}^{(1:t)}$, and sampling locations $\mathbf{x}^{(1:t)}$, the posterior distribution of $\phi(\cdot)$ can be modeled as a Gaussian process with mean $\mu^{(t)}(x)$ and covariance $k^{(t)}(x,x')$, computed as in [27]:

$$\begin{aligned}\mu^{(t)}(x)&=\mu(x;\rho^{(t)})+\mathbf{k}(x;\mathbf{x}^{(1:t)})^{\intercal}\\ &\qquad(\mathbf{K}_{\mathbf{x}^{(1:t)}}(\tau^{(t)})+\sigma^2\mathbf{I})^{-1}(\mathbf{y}^{(1:t)}-\boldsymbol{\mu}^{(1:t)}(\rho^{(t)})),\\ k^{(t)}(x,x')&=k(x,x';\tau^{(t)})\\ &\qquad-\mathbf{k}(x;\mathbf{x}^{(1:t)})^{\intercal}(\mathbf{K}_{\mathbf{x}^{(1:t)}}(\tau^{(t)})+\sigma^2\mathbf{I})^{-1}\mathbf{k}(x';\mathbf{x}^{(1:t)}),\end{aligned} \tag{7}$$

where $\mathbf{k}(x;\mathbf{x}^{(1:t)})\in\mathbb{R}^{Nt}$ is the vector with entries $k(x_i^{(j)},x;\tau^{(t)})$ for $i\in\{1,\dots,N\}$ and $j\in\{1,\dots,t\}$. At each point we can define the standard deviation as

$$\sigma^{(t)}(x)=\sqrt{k^{(t)}(x,x)}. \tag{8}$$

These computations allow the algorithms in Section IV to predict the value of ϕ across the domain.

The described estimation requires a centralized computation of the mean and variance of the Gaussian process. Decentralization of the procedure is left for future work.

IV. ALGORITHM

In this section, we present two different algorithms for the simultaneous estimation and coverage of an initially unknown density function or spatial field. Both algorithms aim to balance the spatial coverage of the estimated density with the decrease in the uncertainty of the estimation. However, the main difference between the two algorithms, hereafter referred to as (i) the Lookahead Estimate Coverage (LEC) algorithm and (ii) the Voronoi Estimate Coverage (VEC) algorithm, is the exploration strategy, i.e., the sampling strategy used to reduce the uncertainty of the model.

We start by defining some useful notation. Let $\phi^{(t)}$ be the estimate of the field ϕ at time t, which we refer to as the *surrogate* density function since it will substitute for the true function in the coverage optimization. We define $\{\beta^{(t)}\}_{t=1}^{T}$ to be a non-negative non-decreasing sequence and $\{\gamma^{(t)}\}_{t=1}^{T}$ to be a non-negative decreasing sequence such that $\gamma^{(t)}\leq 1,\forall t\in\{1,\dots,T\}$. *Optimal coverage positions* refer to a centroidal Voronoi tesselation (CVT) of the estimated field and *optimal estimation positions* as N points where taking measurements will maximally reduce the uncertainty associated with the estimate of the field. Let $V_i^{(t)}$ and $e_i^{(t)}$ be the Voronoi cell and the optimal estimation position of Robot i at time t respectively. Here we omit the dependency of $c_i^{(t)}$ and $V_i^{(t)}$ on $\mathbf{x}^{(t)}$ to simplify the notation.

A. Simultaneous Estimation and Coverage

Solving the coverage problem when the spatial field is unknown requires estimating the density function while distributing robots to monitor the environment with respect to it. We propose the LEC and VEC algorithms to solve this problem and to tackle the drawbacks of the Prediction-Correction Coverage (PCC) algorithm proposed in [32]. In particular, PCC divides the robots into two groups: one robot is dedicated to solely sampling the points where the uncertainty of the estimation is maximal, while the rest of the robots optimize coverage of the estimated density function. At each iteration, the robots take samples once a CVT of the density estimate is achieved.

The PCC approach, therefore, presents two key drawbacks, namely (a) one of the robots does not contribute to the coverage task and (b) the estimation process is suboptimal since the rest of the robots do not aim to reduce the uncertainty of the field. In this paper, we address these drawbacks by making each robot contribute to both learning the field and covering it. We realize this goal by using a

suitable interpolation between the optimal coverage position and optimal estimation position.

1) Lookahead Sampling: The multi-robot team starts with a random initial configuration in the domain, $x^{(0)}$, initialized hyperparameters $\rho^{(0)}, \tau^{(0)}$, Gaussian process mean, $\mu^{(0)}$, and standard deviation, $\sigma^{(0)}$. At each iteration t, the multi-robot team calculates the surrogate of the density function using the update in [32],

$$\phi^{(t)}(q) = \mu^{(t-1)}(q) - \sqrt{\beta^{(t)}}\sigma^{t-1}(q), \quad \forall q \in D, \tag{9}$$

which serves as an approximation of the true density function, ϕ, for the coverage task.

Based on the estimation, using a Lookahead type subroutine [35], LEC finds the N points that would provide the maximum reduction in the estimation uncertainty,

$$\mathbf{x}_e^{(t)} = \left[x_{1,e}^{(t)\intercal}, \ldots, x_{N,e}^{(t)\intercal} \right]^\intercal.$$

Each Robot i is assigned one estimation position, $e_i^{(t)}$, such that $\mathbf{e}^{(t)} = \left[e_1^{(t)\intercal}, \ldots, e_N^{(t)\intercal} \right]^\intercal$ is a permutation of $\mathbf{x}_e^{(t)}$ that minimizes the pairwise distance between each robot, x_i, and the assigned point, $x_{i,e}$.

With respect to the coverage objective, recall that, for a known density function, the optimal spatial allocation is a CVT, which can be attained executing the control law in (3). Analogously, in our algorithm, we optimize the coverage of the surrogate at iteration t, $\phi^{(t)}$, by letting each robot execute

$$\dot{x}_i = \kappa \left(c_i^{(t)} - x_i \right), \quad \kappa > 0,$$

where $c_i^{(t)}$ denotes the center of mass of the Voronoi cell calculated with respect to the surrogate, $\phi^{(t)}$.

We can make the robots concurrently learn the field and cover it by balancing the estimation and the coverage objectives. To this end, the control law

$$\dot{x}_i = \kappa \left((1-\gamma^{(t)}) c_i^{(t)} + \gamma^{(t)} e_i^{(t)} - x_i \right), \quad \kappa > 0 \tag{10}$$

makes each robot evolve towards the interpolated position between $c_i^{(t)}$ and $e_i^{(t)}$, obtaining a trade-off between learning and coverage. The purpose of the decreasing sequence $\{\gamma^{(t)}\}_{t=1}^T$ is to increase the priority given to coverage with increasing number of iterations. Consequently, we choose the sequence such that $\gamma^{(t)} \to 0$ as $t \to \infty$.

The control law in (10) reaches an equilibrium point, where the robots are located at

$$x_i^{(t)} = (1-\gamma) c_i^{(t)} + \gamma e_i^{(t)}, \quad i \in \{1, \ldots, N\}, \tag{11}$$

and take measurements $\mathbf{y}^{(t)}$ according to (4). The hyperparameters $\rho^{(t)}, \tau^{(t)}$ are updated using (6) and the Gaussian process parameters $\mu^{(t)}, \sigma^{(t)}$, needed to calculate the density surrogate, are obtained from (7) and (8).

The pseudo code for LEC is given in Algorithm 1.

Algorithm 1: Lookahead Estimate Coverage (LEC)

Input: Number of robots N, parameter κ, Domain D, sequences $\{\beta^{(t)}\}_{t=1}^T$ and $\{\gamma^{(t)}\}_{t=1}^T$

Initialization;
Random initial positions of robots $\mathbf{x}^{(0)} = [x_1^{(0)\intercal}, \ldots, x_N^{(0)\intercal}]^\intercal$
Calculate initial parameter values $\rho^{(0)}$ and $\tau^{(0)}$
Calculate initial mean $\mu^{(0)}(\cdot)$ and standard deviation $\sigma^{(0)}(\cdot)$
for each iteration $t \in \{1, \ldots, T\}$ **do**
  Calculate the estimate of ϕ for all $x \in D$ as $\phi^{(t)}(x) = \mu^{(t-1)}(x) - \sqrt{\beta^{(t)}}\sigma^{(t-1)}(x)$
  Calculate optimal estimation positions $\mathbf{x}_e^{(t)} = \mathsf{Lookahead}\left(\sigma^{(t-1)}\right)$
  Robots choose estimation positions as $\mathbf{e}^{(t)} = \operatorname{argmin}_{\bar{\mathbf{x}}} \sum_{i=1}^N \|\bar{x}_i - x_i^{(t-1)}\| \; s.t. \; \bar{x}_i \in \mathbf{x}_e^{(t)}$ and $\bar{x}_i \neq \bar{x}_j \; \forall i, j \in \{1, \ldots, N\}$
  while $\exists i \; s.t. \; x_i \neq (1-\gamma) c_i^{(t)} + \gamma e_i^{(t)}$ **do**
    Compute $c_i^{(t)}$ as in (2)
    $\dot{x}_i = \kappa \left((1-\gamma^{(t)}) c_i^{(t)} + \gamma^{(t)} e_i^{(t)} - x_i \right)$
  end
  for each robot $i \in \{1, \ldots, N\}$ **do**
    Obtain the measurement $y_i^{(t)} = \phi(x_i^{(t)}) + \epsilon_i^{(t)}$
  end
  Update parameter values $\rho^{(t)}$ and $\tau^{(t)}$
  Update $\mu^{(t)}(\cdot)$ and $\sigma^{(t)}(\cdot)$
end

2) Voronoi Sampling: In LEC, robots choose the optimal estimation positions, $\mathbf{e}^{(t)}$, in a centralized manner. Alternatively, in VEC, we take advantage of the decentralized nature lent by the partition of the domain in regions of dominance and we restrict the sampling space of each robot to its Voronoi cell, V_i.

Analogously to LEC, in VEC the positions of the team are initialized at random within the domain. The hyperparameters of the Gaussian processes and mean and variance of the surrogate of ϕ are also initialized. At each iteration t, the surrogate function $\phi^{(t)}$ is calculated as in (9). The main difference with LEC is that, in VEC, each Robot i chooses the estimation position $e_i^{(t)}$ as the point with maximum uncertainty within its Voronoi cell, i.e.,

$$e_i^{(t)} = \operatorname*{argmax}_{q \in V_i^{(t-1)}} \sigma^{(t-1)}(q). \tag{12}$$

Executing the control law in (10) leads the multi-robot team to the equilibrium point with the form of (11), where the estimation objectives are given by (12). The samples taken by the robots, $\mathbf{y}^{(t)}$, are then used to update $\rho^{(t)}, \tau^{(t)}$, $\mu^{(t)}, \sigma^{(t)}$, and, consequently, the estimation of the density function, $\phi^{(t+1)}$. The pseudo code for VEC is given in Algorithm 2.

Algorithm 2: Voronoi Estimate Coverage (VEC)

Input: Number of robots N, parameter κ, Domain D, sequences $\{\beta^{(t)}\}_{t=1}^{T}$ and $\{\gamma^{(t)}\}_{t=1}^{T}$

Run lines 1-4 of Algorithm 1.
for each iteration $t \in \{1, \dots, T\}$ **do**
  Run line 6 of Algorithm 1.
  for each robot $i \in \{1, \dots, N\}$ **do**
    Choose estimation position as $e_i^{(t)} = \operatorname{argmax}_{x \in V_i^{(t-1)}} \sigma^{(t-1)}(x)$
  end
  Run lines 9-17 of Algorithm 1.
end

B. Theoretical Results

In this section we provide the theoretical analysis of the performance of the proposed LEC and VEC algorithms. As a metric, we use the additional cost, referred to as *regret*, incurred by the robot team as a consequence of not knowing the field. This can be defined as follows.

Definition 1. *Let $r^{(t)}$ be the regret suffered by robots at iteration t. Then $r^{(t)}$ can be defined as*

$$r^{(t)} = \mathcal{H}(\phi, \mathbf{x}^{(t)}) - \min_{\mathbf{x}} \mathcal{H}(\phi, \mathbf{x}). \tag{13}$$

Let $R^{(T)}$ denote the cumulative regret at iteration T. Then we have $R^{(T)} = \sum_{t=1}^{T} r^{(t)}$. We proceed to upper bound the cumulative regret as follows. We start by stating the additional assumption made in the analysis and a few useful lemmas.

Assumptions. We make the following assumptions.
(A1). *The parameters of the Gaussian process ρ and τ are known.*
(A2). *The domain D is discrete.*
(A3). *The coverage algorithm given in* (3) *for CVT obtains the solution of the optimization problem:* $\mathbf{x} = \operatorname{argmin}_{\bar{\mathbf{x}}} \mathcal{H}(\phi, \bar{\mathbf{x}})$.

The same assumptions are made in [32] for the theoretical results provided for PCC. We refer the readers to [32] for a detailed description of implications of these assumptions.

Recall that all the additive measurement noises $\epsilon_i^{(t)}$ are i.i.d. with $\mathcal{N}(0, \sigma^2)$. Let $I(\phi; \mathbf{y}^{(1:T)})$ be the mutual information between ϕ and its measurements $\mathbf{y}^{(1:T)}$.

Lemma 1. *The information gain for the sample points $\mathbf{y}^{(1:T)}$ satisfies*

$$I\left(\phi(\mathbf{x}^{(1:T)}); \mathbf{y}^{(1:T)}\right) \geq \frac{1}{2} \sum_{t=1}^{T} \log\left(1 + \frac{\left(\max_{q \in \mathbf{x}^{(t)}} \sigma^{(t-1)}(q)\right)^2}{\sigma^2}\right).$$

Proof. The proof directly follows from Lemma 5.3 in [27]. □

Lemma 2. (This is a restatement of Lemma 5.6 in [27]). *Let $\delta \in (0,1)$ and $\beta^{(t)} = 2\log\left(\frac{|D|\pi^{(t)}}{\delta}\right)$, where $\sum_{t \geq 1} \frac{1}{\pi^{(t)}} = 1, \pi^{(t)} > 0$. Then $\forall x \in D$,*

$$|\phi(x) - \mu^{(t-1)}(x)| \leq \sqrt{\beta^{(t)}}\sigma^{(t-1)}(x)$$

holds with probability at least $1 - \delta$.

Now we state the main theoretical result of this paper.

Theorem 1. *Let $d = \max_{q,q' \in D} \|q - q'\|^2$, $A = \int_D \phi(q)\, dq$, and $\eta = \frac{8d^2D^2}{\log(1+\sigma^{-2})}$. Choose $\{\gamma^{(t)}\}_{t=1}^{T}$ such that $\sum_{t=1}^{T} \left(\gamma^{(t)}\right)^2 = o(T)$. Let $\delta \in (0,1)$ and $\beta^{(t)} = 2\log\left(\frac{|D|\pi^{(t)}}{\delta}\right)$, where $\sum_{t \geq 1} \frac{1}{\pi^{(t)}} = 1, \pi^{(t)} > 0$. Let $\xi^{(T)} = \max_{\mathcal{Z} \in D : |\mathcal{Z}| = T} I(\phi_{\mathcal{Z}}; y_{\mathcal{Z}})$. Then with at least probability $1 - \delta$, regret can be upper bounded as*

$$R^{(T)} \leq \sqrt{\eta N T \beta^{(T)} \xi^{(T)}} + o(T).$$

Using $\gamma^{(t)} = 1/t$ regret can be upper bounded as

$$R^{(T)} \leq \sqrt{\eta N T \beta^{(T)} \xi^{(T)}} + dA\frac{\pi^2}{6}.$$

Proof. Here we consider that the high probability event given in Lemma 2 holds. Let $d = \max_{q,q' \in D} \|q - q'\|^2$ and $A = \int_D \phi(q)\, dq$. Recall that for each Robot i we have $x_i^{(t)} = (1 - \gamma^{(t)})c_i^{(t)} + \gamma^{(t)} e_i^{(t)}$. Using (1) we get

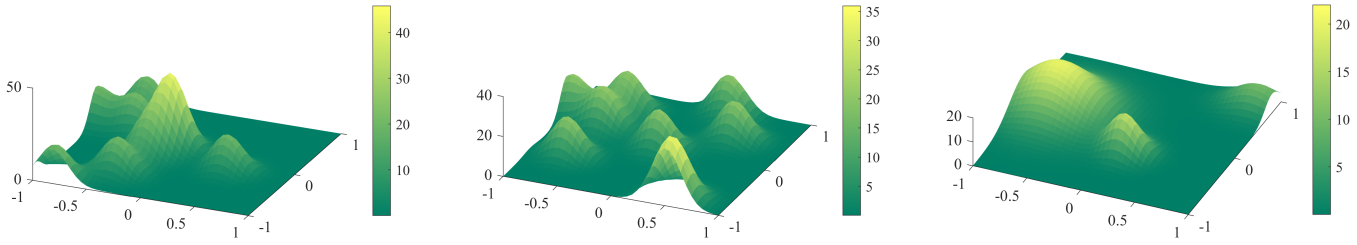
$$\begin{aligned}
\mathcal{H}\left(\phi^{(t)}, \mathbf{x}^{(t)}\right) &= \sum_{i=1}^{N} \int_{V_i^{(t)}} \|q - x_i^{(t)}\|^2 \phi^{(t)}(q)\, dq \\
&\leq \sum_{i=1}^{N} \int_{V_i^{(t)}} \left(\|q - c_i^{(t)}\|^2 + \left(\gamma^{(t)}\right)^2 \|c_i^{(t)} - e_i^{(t)}\|^2\right) \phi^{(t)}(q)\, dq \\
&= \mathcal{H}\left(\phi^{(t)}, \mathbf{c}^{(t)}\right) + \left(\gamma^{(t)}\right)^2 \sum_{i=1}^{N} \int_{V_i^{(t)}} \|c_i^{(t)} - e_i^{(t)}\|^2 \phi^{(t)}(q)\, dq \\
&\leq \mathcal{H}\left(\phi^{(t)}, \mathbf{c}^{(t)}\right) + dA\left(\gamma^{(t)}\right)^2.
\end{aligned} \tag{14}$$

Let $\mathcal{H}^* = \min_{\mathbf{x}} \mathcal{H}(\phi, \mathbf{x})$ and $\mathbf{c}^* = \operatorname{argmin}_x \mathcal{H}(\phi, \mathbf{x})$. From assumption (A3) we have that $\mathcal{H}(\phi^{(t)}, \mathbf{c}^*) \leq \mathcal{H}^*$ for all t. Further we have that $\mathcal{H}(\phi^{(t)}, \mathbf{c}^{(t)}) \leq \mathcal{H}(\phi^{(t)}, \mathbf{c}^*)$. Using (14) we get

$$\mathcal{H}(\phi^{(t)}, \mathbf{x}^{(t)}) - dA\left(\gamma^{(t)}\right)^2 \leq \mathcal{H}(\phi^{(t)}, \mathbf{c}^*) \leq \mathcal{H}^*. \tag{15}$$

From Definition 1 and (15) we have

$$\begin{aligned}
r^{(t)} &= \mathcal{H}\left(\phi, \mathbf{x}^{(t)}\right) - \mathcal{H}^* \\
&\leq \mathcal{H}\left(\phi, \mathbf{x}^{(t)}\right) - \mathcal{H}(\phi^{(t)}, \mathbf{x}^{(t)}) + dA\left(\gamma^{(t)}\right)^2 \\
&\leq 2d|D|\sqrt{\beta^{(t)}} \max_{q \in \mathbf{x}^{(t)}} \sigma^{(t-1)}(q) + dA\left(\gamma^{(t)}\right)^2.
\end{aligned} \tag{16}$$



(a) ϕ_1. (b) ϕ_2. (c) ϕ_3.

Fig. 1. Density functions used for the simulations and experiments. ϕ_1 and ϕ_2, used for the simulations, are a sum of 9 Gaussian bivariate distributions with equal variance and random means within the domain. Robotic experiments were carried out with ϕ_3, a sum of 3 Gaussians with different variances.

TABLE I

NORMALIZED REGRET AND NORMALIZED SURROGATE ERROR AT $T = 20$ ITERATIONS

	density, ϕ	N	Benevento [32]		Luo [25]		LEC		VEC	
			$r^{(T)}$	$\Delta^{(t)}$	$r^{(T)}$	$\Delta^{(t)}$	$r^{(T)}$	$\Delta^{(t)}$	$r^{(T)}$	$\Delta^{(t)}$
Simulation 1	ϕ_1	4	0.2847	0.3256	0.0705	0.3659	0.0374	0.2327	0.0326	0.2226
Simulation 2		7	0.0256	0.1999	0.0500	0.2702	0.0584	0.0753	0.0281	0.1622
Simulation 3	ϕ_2	4	0.1517	0.3268	0.4116	0.6686	0.2264	0.3001	0.1000	0.2999
Simulation 4		7	0.0817	0.1483	0.0610	0.2983	0.0118	0.0651	0.0185	0.1254

Since $\{\beta^{(t)}\}_{t=1}^{T}$ is non-decreasing we have

$$4d^2D^2\beta^{(t)}\left(\max_{q\in\mathbf{x}^{(t)}}\sigma^{(t-1)}(q)\right)^2 \leq \eta\beta^{(T)}\frac{1}{2}\log\left(1+\frac{\left(\max_{q\in\mathbf{x}^{(t)}}\sigma^{(t-1)}(q)\right)^2}{\sigma^2}\right) \quad (17)$$

where $\eta = \frac{8d^2D^2}{\log(1+\sigma^{-2})}$. Further we have

$$I\left(\phi(\mathbf{x}^{(1:T)}),\mathbf{y}^{(1:T)}\right) \leq N \max_{\mathcal{Z}\in D:|\mathcal{Z}|=T} I\left(\phi_{\mathcal{Z}};y_{\mathcal{Z}}\right). \quad (18)$$

Let $\xi^{(T)} = \max_{\mathcal{Z}\in D:|\mathcal{Z}|=T} I\left(\phi_{\mathcal{Z}};\mathbf{y}_{\mathcal{Z}}\right)$. Then from Lemma 1, (16), (17) and (18) we have

$$\sum_{t=1}^{T} r^{(t)} \leq \sqrt{\eta NT\beta^{(T)}\xi^{(T)}} + dA\sum_{t=1}^{T}\left(\gamma^{(t)}\right)^2. \quad (19)$$

Choose $\{\gamma^{(t)}\}_{t=1}^{T}$ such that $\sum_{t=1}^{T}\left(\gamma^{(t)}\right)^2 = o(T)$. Then from (19) we get

$$R^{(T)} \leq \sqrt{\eta NT\beta^{(T)}\xi^{(T)}} + o(T).$$

Substituting $\gamma^{(t)} = \frac{1}{t}$ into (19) we get

$$R^{(T)} \leq \sqrt{\eta NT\beta^{(T)}\xi^{(T)}} + dA\frac{\pi^2}{6}.$$

This concludes the proof of Theorem 1. □

Remark 1. *Note that we can choose a sequence $\{\beta^{(t)}\}_{t=1}^{T}$ such that $\beta^{(T)} = O(\log T)$. Further from [27] Theorem 8 we have that $\xi^{(T)}$ scales sublinearly with many kernels. Thus we see that our algorithms accumulates only a logarithmic regret.*

V. SIMULATIONS AND EXPERIMENTAL RESULTS

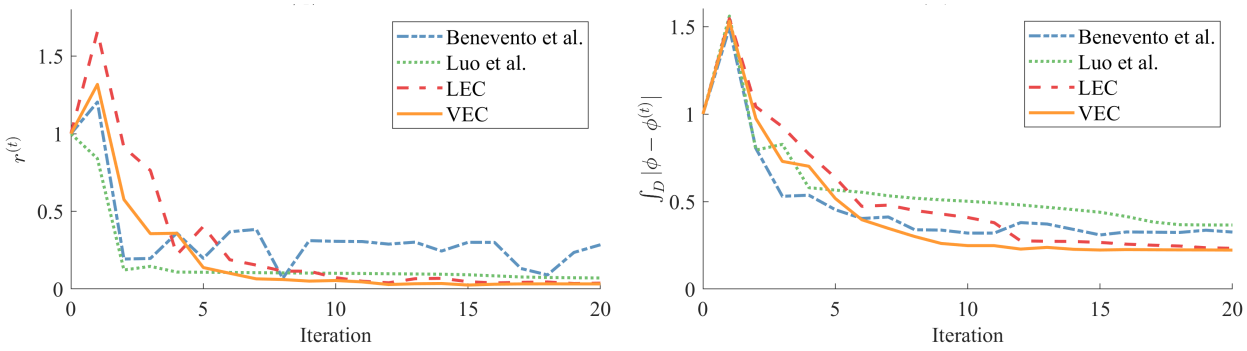
The proposed algorithms are implemented in simulation as well as on a team of real robots. The first part of this section includes simulations where the performance of both algorithms is compared with other approaches from the literature. The second part includes experiments on a real robotic platform that were conducted to verify the performance of the algorithms and evaluate the potential differences between the Lookahead and Voronoi sampling strategies.

A. Simulations

The LEC and VEC algorithms are implemented in simulation to provide a comprehensive comparison against algorithms that also tackle simultaneous estimation and coverage of spatial fields. To this end, we focus on approaches where the control law is based on the continuous version of Lloyd's algorithm in (3) executed with respect to a surrogate function. With this criterion, the algorithms selected for the comparison are the ones presented in [25] and [32], hereafter referred to as Luo *et al.* and Benevento *et al.*, respectively.

Each algorithm is simulated in the scenarios described in Table I over $T = 20$ iterations. The simulations include teams of four and seven robots and two different densities, ϕ_1 and ϕ_2, defined in a 2m×2m domain. These density functions are calculated as the sum of nine Gaussians with identical variance with random means within the domain boundaries and are depicted in Figs. 1a and 1b. For each scenario in Table I, the initial position of the robot team is drawn at random from a uniform distribution in the domain. In the LEC and VEC algorithms, the decreasing sequence that balances the execution of coverage and estimation is defined as $\{\gamma^{(t)}\} = \{1/t\}_{t=1}^{T}$.

We use two different metrics to evaluate the performance of the algorithms: the regret, $r^{(t)}$, defined in (13), and the



(a) Regret, $r^{(t)}$. (b) Density error, $\Delta^{(t)}$

Fig. 2. Evolution of (a) the regret and (b) the density error for Simulation 1. The initial peak at Iteration 1 in the regret is caused by a pure sampling strategy in LEC and VEC, since the chosen decreasing sequence gives $\gamma^{(1)} = 1$. While the initial regret is slightly bigger for LEC and VEC, after 10 iterations, they outperform Luo *et al.* and Benevento *et al.*, achieving a final lower regret. The evolution of the regret can be further understood by considering the density error, where it can be observed that the estimation of the density function becomes better for the case of the LEC and the VEC algorithms as $t = 10$ iterations is surpassed. A better final estimation of the density function leads to a lower value of regret for the proposed algorithms.

error between the surrogate and the true density,

$$\Delta^{(t)} = \int_D |\phi(q) - \phi^{(t)}(q)|\, dq,$$

which measures the quality of the estimation.

The evolution of the regret for Simulation 1 is depicted in Fig. 2a. We can observe how the proposed LEC and VEC algorithms present an initial increase in the regret, due to the initial prioritization of exploration ($\gamma^{(1)} = 1$), which results in the robots sampling around areas of high uncertainty. As can be observed in the companion video, at the initial stages of the algorithm, these points often correspond to boundary points that lend poor coverage performance. However, once the samples are enough to obtain a good estimation of the density function, the regret of LEC and VEC becomes lower than the other two algorithms. This is consistent with the values of final regret observed in the other simulations in Table I, where the LEC and VEC algorithms generally show a lower value of regret. For the cases where the regret is not strictly lower, its value is comparable to the lower value between Luo *et al.* and Benevento *et al*.

The evolution of the density error for Simulation 1 is shown in Fig. 2b where, after an initial increase in the error, the estimation decreases for all the algorithms. However, the benefits of the LEC and the VEC are highlighed by this metric, where the differences between the quality of field estimation become more prominent for the proposed approaches, which yield better surrogate functions. This is consistent with the final values for the density error in Table I, where the proposed algorithms outperform the other two algorithms in all four scenarios.

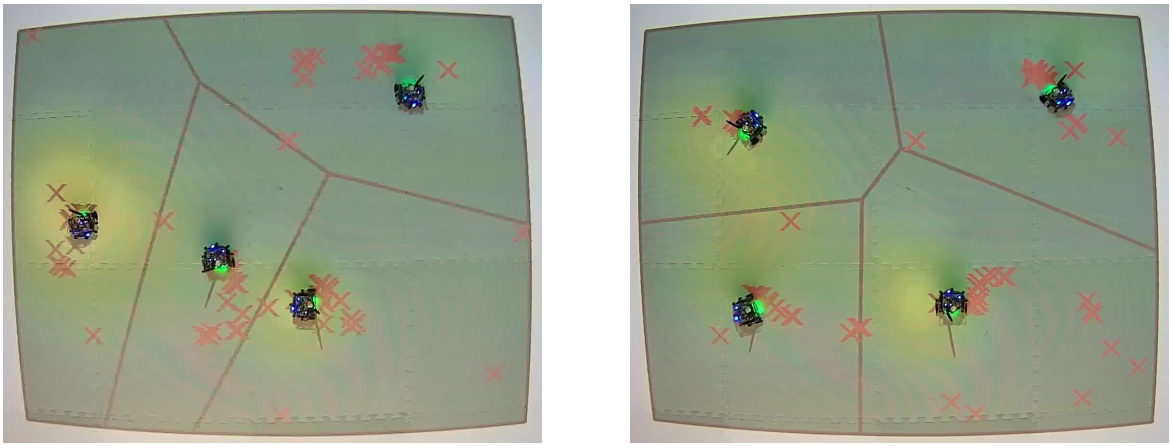
Given the satisfactory results observed in these simulations under ideal assumptions such as single integrator dynamics, without delays or actuator limits, in next section we limit the experimental implementation on real robots to only the proposed algorithms, LEC and VEC.

B. Robotic Experiments

The LEC and VEC algorithms were implemented on a team of four mobile robots on the Robotarium [36], a remotely accessible swarm robotics testbed where experiments can be uploaded via web. The robots have a differential-drive kinematic configuration and, thus, their movement can be described using unicycle dynamics, i.e., $\dot{x}_i = [v_i \cos\theta_i, v_i \sin\theta_i]^\intercal$, $\dot{\theta}_i = \omega_i$, where (v_i, ω_i) denote the linear and angular velocity of robot i and θ_i, its heading. At each control iteration, the control script receives the poses of the robots, does the calculations, and outputs the desired linear and angular velocity for all the robots. Given that Algorithms 1 and 2 consider single integrator dynamics, we transformed the control laws into unicycle dynamics using a diffeomorphism analogous to the one in [37].

In the experiment, four robots were tasked with learning and covering the density ϕ_3 (Fig. 1c). As can be observed in the snapshots in Fig. 3, both in the case of LEC and VEC the surrogate function learned by the robots matches closely the actual density function, ϕ_3. This is supported by the density error plot in Fig. 4, where we can observe how the surrogate error in both algorithms converges to a very similar value. Despite the similarity between the learned surrogate functions, however, the final spatial allocations of the multi-robot team in Fig. 3 differ slightly due to the different trajectories adopted by the multi-robot team during the execution of the environment, which led the team to converge to two different, albeit close, local minima.

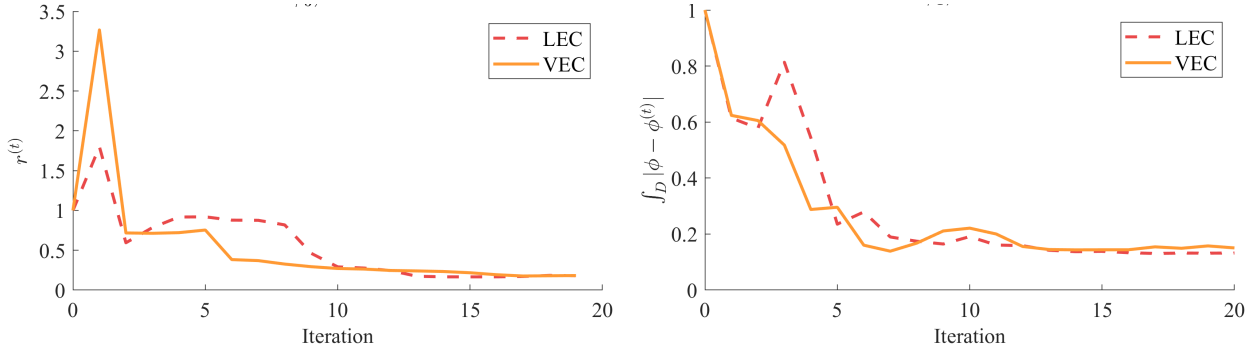
The regret of both LEC and VEC algorithms in the different iterations is also shown in Fig. 4. Analogously to what we observed in the simulations in Fig. 2, the balancing factor between the learning and the coverage objectives causes an initial increase in the regret, which corresponds to the maxima observed at Iteration 1. However, after this first exploration step, the regret rapidly decreases for both methods as the robots improve the estimation of the density. After approximately 10 iterations, the robots have a fairly good estimation of the density function and focus mainly



(a) Lookahead Estimate Coverage (LEC) Algorithm.

(b) Voronoi Estimate Coverage (VEC) Algorithm.

Fig. 3. Final spatial configurations of the experiments run on a team of four robots learning and covering density ϕ_3, as shown in Fig. 1c. The learned surrogate function is represented by a heat map, with higher values of the density being depicted in yellow and lower values in deep green. Within the domain, the straight lines represent the boundaries of the Voronoi cells corresponding to the regions of dominance of the different robots. The red $\times$ show the points where the robots have taken a sample in the domain according to lines 8 and 5 in Algorithms 1 and 2, respectively.



(a) Regret, $r^{(t)}$.

(b) Density error, $\Delta^{(t)}$.

Fig. 4. Evolution of (a) the regret and (b) the density error for the robotic experiment, where a team of four robots is tasked with exploring and covering ϕ_3. Similarly to what was observed in the simulations, the regret presents an initial increase in the first iteration of the LEC and VEC algorithms due to a pure learning strategy. After this increase, the regret rapidly decreases for both algorithms, stabilizing approximately after 10 iterations. The evolution of the error between the surrogate and the true density function presents similar results for the LEC and the VEC algorithms, suggesting that limiting the sampling space in the case of the second algorithm does not affect significantly the learning of the density function.

on the coverage task, given that the sequence $\{\gamma^{(t)}\}$ goes to almost zero at $T = 20$. In the snapshot of the final configuration of the experiments in Fig. 3, we can see how these iterations correspond to the sampling points (marked as red $\times$) that are close to the final positions of the robots.

VI. CONCLUSIONS

In this paper, we introduced two algorithms for a team of robots to simultaneously learn and cover an initially unknown density function. The crux of the approach lies in executing a standard optimal coverage control law with respect to a surrogate function of the true density function, which is iteratively refined as the robots explore and collect samples over the domain. The novelty of this approach is based on the sampling (exploration) strategies associated with the two algorithms, which determine which points to visit based on their associated uncertainty. The algorithms differ on the sampling space available to each robot and the assignment of the desired sampling points to the individual robots in the team. Theoretical guarantees were presented in terms of regret for the algorithms. Their performance was compared to other methods in simulation, showing improvement both in terms of regret and density function estimation, with the proposed algorithms rendering a close estimation of the true density function. Further, limiting each robot's sampling space in the case of the Voronoi estimation did not show a significant performance drop with respect to the centralized sampling based on lookahead methods, which is a promising in the context of decentralization of the algorithm. Both algorithms were evaluated on a team of real robots with satisfactory results, showing that delays and actuator limits did not affect the performance of the algorithms.

ACKNOWLEDGMENT

M. S. thanks Dr. Luis Guerrero-Bonilla and Yousef Emam for their invaluable help with the Robotarium experiments.

REFERENCES

[1] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 2, pp. 243–255, Apr. 2004.

[2] F. Bullo, J. Cortés, and S. Martínez, *Distributed Control of Robotic Networks: A Mathematical Approach to Motion Coordination Algorithms*. Princeton University Press, 2009.

[3] M. Schwager, J. McLurkin, and D. Rus, "Distributed coverage control with sensory feedback for networked robots," in *Proceedings of Robotics: Science and Systems*, 2006.

[4] L. C. A. Pimenta, V. Kumar, R. C. Mesquita, and G. A. S. Pereira, "Sensing and coverage for a network of heterogeneous robots," in *2008 47th IEEE Conference on Decision and Control*, 2008, pp. 3947–3952.

[5] G. Kantor, S. Singh, R. Peterson, D. Rus, A. Das, V. Kumar, G. Pereira, and J. Spletzer, *Distributed Search and Rescue with Robot and Sensor Teams*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 529–538.

[6] A. Barrientos, J. Colorado, J. d. Cerro, A. Martinez, C. Rossi, D. Sanz, and J. Valente, "Aerial remote sensing in agriculture: A practical approach to area coverage and path planning for fleets of mini aerial robots," *Journal of Field Robotics*, vol. 28, no. 5, pp. 667–689, 2011.

[7] M. Dunbabin and L. Marques, "Robots for environmental monitoring: Significant advancements and applications," *IEEE Robotics Automation Magazine*, vol. 19, no. 1, pp. 24–39, 2012.

[8] N. E. Leonard, D. A. Paley, F. Lekien, R. Sepulchre, D. M. Fratantoni, and R. E. Davis, "Collective motion, sensor networks, and ocean sampling," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 48–74, 2007.

[9] N. E. Leonard, "Cooperative vehicle environmental monitoring," in *Springer Handbook of Ocean Engineering*, M. R. Dhanak and N. I. Xiros, Eds. Springer International Publishing, 2016, pp. 441–458.

[10] A. Dhariwal, G. Sukhatme, and A. Requicha, "Bacterium-inspired robots for environmental monitoring," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, vol. 2, 2004, pp. 1436–1443 Vol.2.

[11] A. Kwok and S. Martinez, "Deployment algorithms for a power-constrained mobile sensor network," in *2008 IEEE International Conference on Robotics and Automation*, 2008, pp. 140–145.

[12] M. Santos and M. Egerstedt, "Coverage control for multi-robot teams with heterogeneous sensing capabilities using limited communications*," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 5313–5319.

[13] A. Sadeghi and S. L. Smith, "Coverage control for multiple event types with heterogeneous robots," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 3377–3383.

[14] F. Lekien and N. E. Leonard, "Nonuniform coverage and cartograms," *SIAM Journal on Control and Optimization*, vol. 48, no. 1, pp. 351–372, 2009.

[15] L. C. A. Pimenta, M. Schwager, Q. Lindsey, V. Kumar, D. Rus, R. C. Mesquita, and G. A. S. Pereira, *Simultaneous Coverage and Tracking (SCAT) of Moving Targets with Robot Networks*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 85–99.

[16] S. G. Lee, Y. Diaz-Mercado, and M. Egerstedt, "Multirobot control using time-varying density functions," *IEEE Transactions on Robotics*, vol. 31, no. 2, pp. 489–493, 2015.

[17] S. Lloyd, "Least squares quantization in pcm," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.

[18] J. Cortes and M. Egerstedt, "Coordinated control of multi-robot systems: A survey," *SICE Journal of Control, Measurement, and System Integration*, vol. 10, no. 6, pp. 495–503, 2017.

[19] A. Singh, A. Krause, and W. J. Kaiser, "Nonmyopic adaptive informative path planning for multiple robots," in *Proceedings of the 21st International Jont Conference on Artifical Intelligence*, ser. IJCAI'09. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2009, p. 1843–1850.

[20] F. Zhang and N. E. Leonard, "Cooperative filters and control for cooperative exploration," *IEEE Transactions on Automatic Control*, vol. 55, no. 3, pp. 650–663, 2010.

[21] C. Peterson and D. A. Paley, "Distributed estimation for motion coordination in an unknown spatially varying flowfields," *Journal of Guidance, Control, and Dynamics*, vol. 36, no. 3, p. 894–898, 2013.

[22] G. A. Hollinger and G. S. Sukhatme, "Sampling-based robotic information gathering algorithms," *The International Journal of Robotics Research*, vol. 33, no. 9, pp. 1271–1287, 2014.

[23] G. Notomista and M. Egerstedt, "Communication constrained distributed spatial field estimation using mobile sensor networks," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 9642–9649, 2020.

[24] D. A. Paley and A. Wolek, "Mobile sensor networks and control: Adaptive sampling of spatiotemporal processes," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, p. 91–114, 2020.

[25] W. Luo and K. Sycara, "Adaptive sampling and online learning in multi-robot sensor coverage with mixture of gaussian processes," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 6359–6364.

[26] W. Luo, C. Nam, G. Kantor, and K. Sycara, "Distributed environmental modeling and adaptive sampling for multi-robot sensor coverage," in *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, 2019, pp. 1488–1496.

[27] N. Srinivas, A. Krause, S. Kakade, and M. Seeger, "Gaussian process optimization in the bandit setting: No regret and experimental design," in *Proceedings of the 27th International Conference on Machine Learning*, ser. ICML'10. USA: Omnipress, 2010, pp. 1015–1022.

[28] M. Schwager, D. Rus, and J.-J. Slotine, "Decentralized, adaptive coverage control for networked robots," *The International Journal of Robotics Research*, vol. 28, no. 3, pp. 357–375, 2009.

[29] M. Schwager, M. P. Vitus, D. Rus, and C. J. Tomlin, "Robust adaptive coverage for robotic sensor networks," in *Robotics Research*. Springer, 2017, pp. 437–454.

[30] A. Carron, M. Todescato, R. Carli, L. Schenato, and G. Pillonetto, "Multi-agents adaptive estimation and coverage control using gaussian regression," in *2015 European Control Conference (ECC)*, 2015, pp. 2490–2495.

[31] L. Wei, A. McDonald, and V. Srivastava, "Multi-robot gaussian process estimation and coverage: A deterministic sequencing algorithm and regret analysis," in *2021 International Conference on Robotics and Automation (ICRA)*, Xi'an China, May 2021.

[32] A. Benevento, M. Santos, G. Notarstefano, K. Paynabar, M. Bloch, and M. Egerstedt, "Multi-robot coordination for estimation and coverage of unknown spatial fields," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 7740–7746.

[33] Q. Du, V. Faber, and M. Gunzburger, "Centroidal voronoi tessellations: Applications and algorithms," *SIAM Rev.*, vol. 41, no. 4, pp. 637–676, Dec. 1999.

[34] C. E. Rasmussen, "Gaussian processes in machine learning," in *Summer School on Machine Learning*. Springer, 2003, pp. 63–71.

[35] C. Chevalier, "Fast uncertainty reduction strategies relying on gaussian process models," Ph.D. dissertation, Universität Bern, 2013.

[36] S. Wilson, P. Glotfelter, L. Wang, S. Mayya, G. Notomista, M. Mote, and M. Egerstedt, "The robotarium: Globally impactful opportunities, challenges, and lessons learned in remote-access, distributed control of multirobot systems," *IEEE Control Systems Magazine*, vol. 40, no. 1, pp. 26–44, 2020.

[37] R. Olfati-Saber, "Near-identity diffeomorphisms and exponential /spl epsi/-tracking and /spl epsi/-stabilization of first-order nonholonomic se(2) vehicles," in *Proceedings of the 2002 American Control Conference (IEEE Cat. No.CH37301)*, vol. 6, 2002, pp. 4690–4695 vol.6.